



Lefol, D., Bull, DR., & Canagarajah, CN. (2006). Mode refinement algorithm for H.264 intra frame requantization. In *2006 IEEE International Symposium on Circuits and Systems, 2006 (ISCAS 2006) Kos, Greece* (pp. 4459 - 4462). Institute of Electrical and Electronics Engineers (IEEE).  
<https://doi.org/10.1109/ISCAS.2006.1693619>

Peer reviewed version

Link to published version (if available):  
[10.1109/ISCAS.2006.1693619](https://doi.org/10.1109/ISCAS.2006.1693619)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

# Mode Refinement Algorithm for H.264 Intra Frame Requantization

Damien Lefol, Dave Bull and Nishan Canagarajah

Department of Electrical and Electronic Engineering

University of Bristol

Bristol, U.K.

Damien.lefol@bristol.ac.uk

**Abstract** – The latest video coding standard H.264 has been recently approved and has already been adopted for numerous applications including HD-DVD and satellite broadcast. To allow interconnectivity between different applications using H.264, transcoding will be a key factor. When requantizing a bitstream the incoming coding decisions are usually kept unchanged to reduce the complexity, but it can have a major impact on the coding efficiency. This paper proposes a novel algorithm for mode refinement of intra prediction in the case of requantization of H.264 intra frames. The proposed approach gives a comparable quality to a full search for a fraction of its complexity by exploiting the statistical properties of the mode distribution.

## I. INTRODUCTION

The new standard H.264 [1] is already successful thanks to the variety of scenarios that it can cover and the high quality of video it can deliver even at low bitrates. Recently it has been provisionally approved as one of the standards for HD-DVD and many broadcasters plan to use it to deliver satellite video.

A large amount of research is ongoing in H.264 which is thought to replace MPEG-2 in the coming years. The applications using H.264 will range from multimedia content delivery on mobile handset to High Definition television broadcasting. To allow such diversity in the video broadcasting, it will be necessary to have means of adapting the video to the distribution channel. One solution would be to store only the highest quality bitstream on the server side and to transcode the bitstream depending on the customers needs.

Many algorithms have been developed for the requantization of video in the last decade. Some of those algorithms, such as the Cascaded Pixel Domain Transcoder (CPDT) [2] and the Fast Pixel Domain Transcoder (FPDT) [3], [4], have been used successfully in many practical applications [5], [6]. It is possible to adapt those algorithms with some changes to the new H.264 standard.

As demonstrated in previous work [7], the FPDT algorithm cannot be used with H.264 bitstream, and the CPDT quality can be significantly lower than a full decode and reencode. This difference is mainly due to the large number of new tools introduced by H.264. The compression efficiency of this new standard is maximal only when all modes are used. When requantizing H.264 bitstream with CPDT, the encoding decisions of the incoming bitstream are kept to reduce the complexity. This implies that the transcoded video uses sub-optimal encoding parameters. A mode refinement algorithm is needed to improve the coding efficiency of the transcoder while reusing as much of the incoming information as possible. The mode refinement algorithm presented in this paper allows to choose the best prediction mode without having to do a full search in the case of intra frame requantization. The simulation results show that it is possible to have a quality close to a full search while saving up to 95% of the complexity.

Section 2 of this paper will give a quick overview of the requantization algorithm used. The limitation of this algorithm and the proposed mode refinement algorithm are described in section 3 together with different possible tuning to balance quality and complexity. Simulation results are given in Section 4. Finally, section 5 concludes the paper.

## II. REQUANTIZATION FOR H.264

The requantization algorithm used in this paper is a Cascaded Pixel Domain Transcoder (CPDT) adapted to H.264 as described in [7]. The transcoding complexity is kept low by reusing the encoding information from the incoming bitstream. This approach yields good results when the transcoded bitrate is close to the original bitrate, but for large differences the quality drops. This is due to a sub-optimal use of H.264 encoding macroblock (MB) modes.

Figure 1, obtained using the experimental setup described in section 4, shows that for intra frame, in a high bitrate video, intra 4x4 modes (I\_4x4) will be predominant

---

The work is funded by 3CRL ROAM4G.

whereas in a low bitrate video intra 16x16 (I\_16x16) can be nearly as frequent.

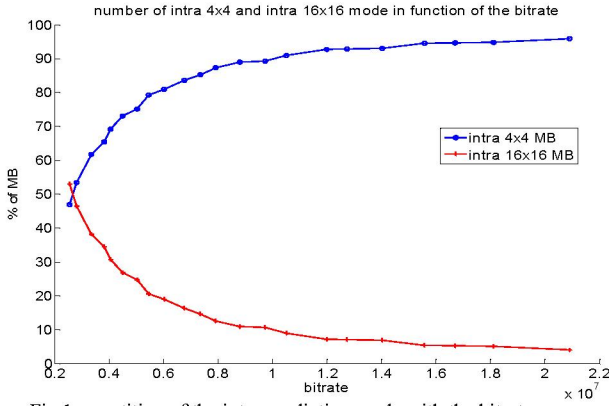


Fig.1 repartition of the intra prediction mode with the bitrate

The increase of bitrate due to this loss of compression efficiency can be as high as 60% when a very high bitrate needs to be transcoded to lower bitrates (Cf. figure 4). To compensate for this loss it is necessary to change at least some of the incoming encoding decision. The algorithm presented below allows to decide when to refine an incoming macroblock or not.

### III. MODE REFINEMENT ALGORITHM

One of the problems of mode refinement is to decide when to refine a macroblock or not since this could lead to poor quality or higher complexity. The statistics obtained from bitstream requantized at different bitrates show that intra 4x4 becomes intra 16x16 when the bitrate decreases. However this requires knowledge of which intra 4x4 become intra 16x16 and what should be done with the remaining intra 4x4. One option is to consider all intra 4x4 but this will increase the complexity. On the other hand when the quantization parameter (QP) increases the cost of encoding intra prediction direction for 4x4 MB can be relatively high. The intra 4x4 mode predicted from its neighbors as most probable is then chosen more often as it will save bits thanks to the intra mode predictive encoding. Failing to use this most probable mode can incur an increase of 25% of the MB headers. The number of mode refinement needed increases with the difference of bitrate between the input and the output of the transcoder.

If  $mb\_type = I\_4x4$ , then

$$\begin{cases} mb\_size < max\_mb\_size \times t_{16} \Rightarrow I\_16x16 \text{ refinement} \\ mb\_size > max\_mb\_size \times t_4 \Rightarrow I\_4x4 \text{ refinement} \end{cases} \quad (1)$$

else if  $mb\_type = I\_16x16$ , then

no mode refinement

The approach chosen to solve this problem is to use two variable thresholds. The first one, called  $t_{16}$ , is used to decide if a I\_16x16 mode should be tried and the second threshold, called  $t_4$ , for I\_4x4 modes. Note that  $t_4$  and  $t_{16}$

evolve in opposite direction. The mode refinement decision is taken at a MB level using the macroblock size (in bits),  $mb\_size$ , as specified in equation 1.

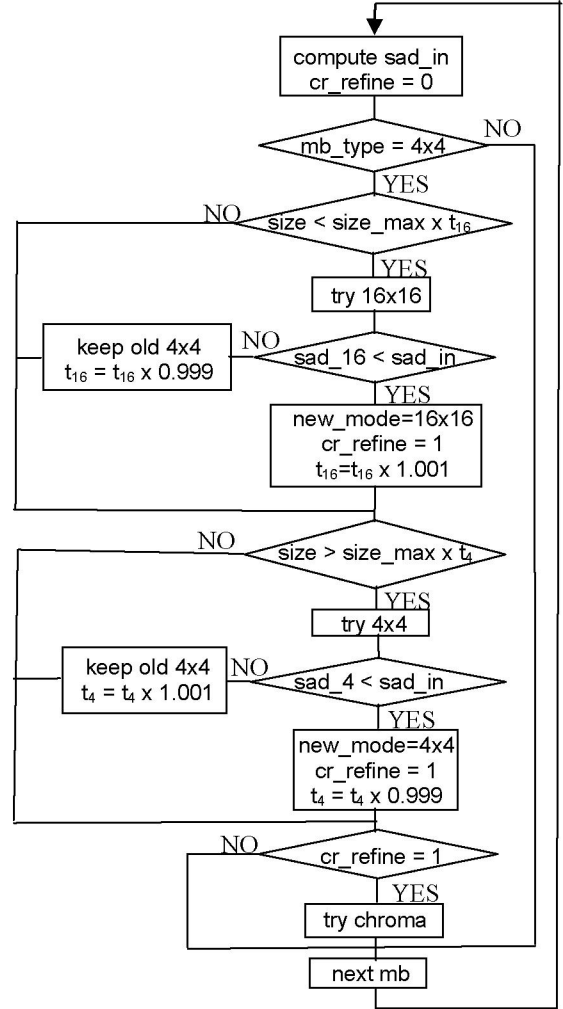


Fig. 2 Flowchart of the threshold mode refinement algorithm

Only incoming MB encoded in 4x4 are concerned by the mode refinement because if the encoder used a 16x16 mode in the high bitrate video it is nearly sure that the same mode will be kept at a lower bitrate. If the mode refinement tried is not better than the original mode (using a sum of absolute difference or SAD comparison) the original mode is kept. The threshold  $t_{16}$  is then decreased and  $t_4$  increased by 0.1% to reduce the chances to try a refinement for the next MB, thus keeping the complexity as low as possible. On the other hand if the mode refinement tried is better, the new mode is kept and the corresponding threshold is modified by 0.1% so that mode refinement is more likely for the next MB. This adaptation of the threshold is done on a macroblock basis because the amount of refinement needed is closely linked to the local video properties. Chroma refinement is turn on if any luma mode refinement tried for this MB has been kept, otherwise no mode refinement is done for the

chroma component of the current MB. The flowchart of our threshold mode refinement algorithm is given in figure 2.

The initial value of the thresholds is a function of the input and output quantization parameters (Cf. equation 2). If  $QP_2$  is smaller than  $QP_1$ , no mode refinement will be tried since this would be done on a noisy video and thus give worse results (Cf figure 7). If  $QP_2$  is higher than  $QP_1$  an initial threshold value is set depending on the difference of QP ( $\Delta QP$ ). The smaller  $\Delta QP$  is, the higher the threshold will be, so less mode refinements are done. The initial values of thresholds have been chosen empirically by trying to match the result given by a full search process.

$$\begin{cases} QP_1 \geq QP_2 \Rightarrow t_4 = 1 \text{ and } t_{16} = 0 \\ QP_1 < QP_2 \Rightarrow t_4 = x[\Delta QP] \text{ and } t_{16} = y[\Delta QP] \end{cases} \quad (2)$$

with  $x, y \in [0,1]$  and  $\Delta QP = QP_2 - QP_1$

#### IV. RESULTS

The simulations were done by transcoding a video sequence composed of three concatenated standard definition sequences (SD 720x576). The first 5 frames are from "Pedestrian", then 5 frames from "Tractor" and the last 5 frames are from "Toys". Those frames are separated by 15 frames in the original sequence so as to represent a typical encoding situation with an intra frame every half a second. The first sequence contains multiple occlusions, the second a tracking camera and high texture and the third, complex motions and uniform areas.

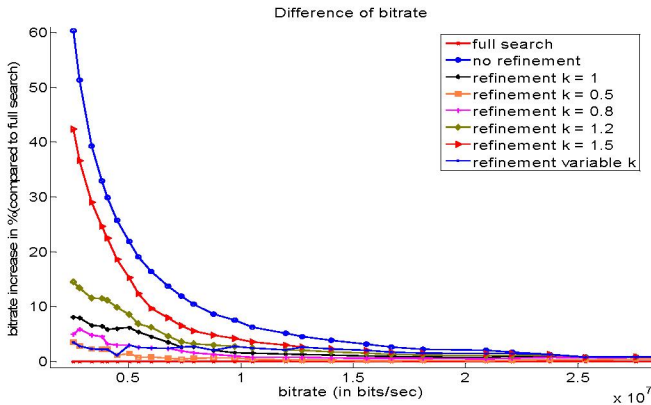


Fig. 3 Bitrate difference for different mode refinement thresholds

The bitstream has been encoded at 30 frames per second with a group of picture containing only intra frames. The original sequence is encoded in H.264 using the reference software JM8.5 at a QP of 10. This sequence is then transcoded at different bitrates using our transcoder. This simulation corresponds to an incoming video at a bitrate of 42.68 Mbps. This is rather high since no inter frames were used in the simulation.

The mode refinement algorithm uses thresholds defined empirically by fitting the statistical distribution of the modes

against the output bitrate wanted. The thresholds values of each possible QP were then optimized to reduce the number of mode refinement tried but not kept. This first approach was refined by applying multiplicative coefficients  $k$ , included between 0.5 and 1.5, to the threshold values so as to have a finer approximation of the optimum thresholds. Figure 3 and 4 shows the results obtained from those simulations. The comparisons are always done with the full search corresponding to 100% of complexity and 0% of bitrate increase.

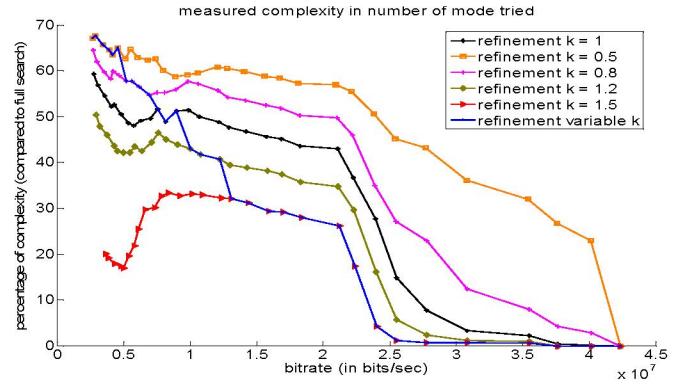


Fig. 4 Complexity comparison for different mode refinement thresholds

Figure 3 shows that doing mode refinement is highly beneficial for the output bitrate of the transcoder. When compared to transcoding without refinement we obtain a gain of up to 55% in terms of bitrate at similar quality. Actually the mode refinement gives result comparable to the full search. Figure 4 shows the complexity in terms of number of modes tried for the developed algorithms. It can be seen that for bitrate close to the original one, there is no increase in complexity; it then increases as more MBs are tried for refinement. It also shows clearly that the balance between complexity and quality of the mode refinement algorithm can be modified gradually by using different values of  $k$ .

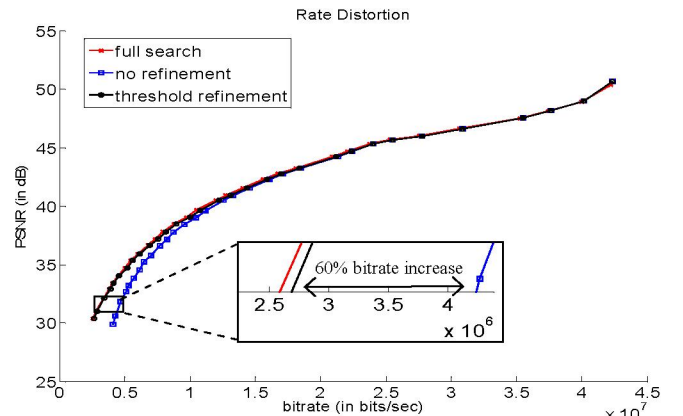


Fig. 5 Rate distortion comparison for an input sequence at QP 10



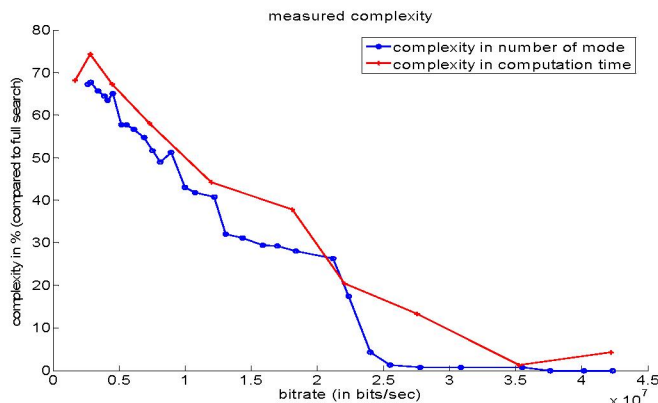


Fig. 6. Complexity of the mode refinement algorithm (input at QP10)

From the simulation done with different multiplicative coefficients  $k$ , we decided which  $k$  to use depending on the difference between input and output bitrate. This approach, called variable  $k$  in the previous graph, is designed by threshold refinement algorithm in the rest of this paper. Figure 5 gives the rate distortion obtained. Figure 6 shows the corresponding complexity in number of modes tried and time spent on mode refinement by the transcoder. It shows that approximating the complexity of the mode refinement algorithm by the number of modes tried is quite close to the reality. Figure 6 shows clearly that when the bitrate of the output bitstream is more than half the input bitrate, the complexity of our algorithm is less than 25% of full search. Even in the case of a bitrate reduction of 85%, the complexity is still kept under 50% compared to full search. In the worst case scenario the complexity is still 25% lower for the same quality.

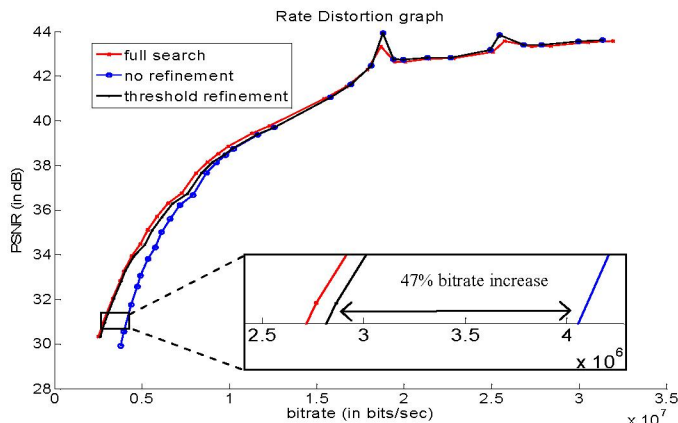


Fig. 7 Rate distortion comparison for an input sequence at QP 20

Figure 7 and Figure 8 shows the validity of those results for a different input bitrate. The transcoder input is now at 18.8 Mbps (QP 20), and the transcoder gives an output ranging from twice to a tenth of the input bitrate. The two peaks on the RD curve are due to matching QPs when trying to transcode at a bitrate higher than the input one.

The mode refinement approach presented here is simple and efficient. It could be easily adapted for mode refinement

in inter frames and linked to motion vector refinement algorithm previous developed such as [8].

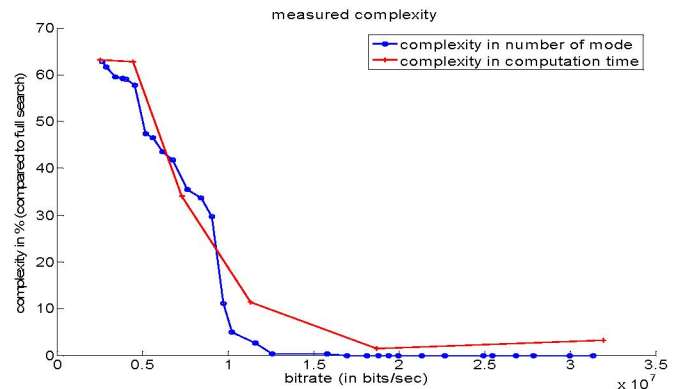


Fig. 8. Complexity of the mode refinement algorithm (input at QP20)

## V. CONCLUSION

H.264 provides an efficient compression standard for video coding, but when requantizing H.264 bitstream its efficiency can be seriously limited if using non optimal macroblock mode and intra prediction mode. When the incoming coding information is reused without mode refinement, the bitrate increase can be as high as 60% compared to a full recode. The threshold mode refinement algorithm presented here provides an efficient tool to keep the quality and bitrate of the transcoded intra frames at its maximum while reducing the complexity by at least 25% and up to 95% compared to a full search approach.

## REFERENCES

- [1] ISO, "ISO/IEC 14496-10 AVC | ITU-T Rec. H.264 International Standard for Advanced Video Coding," 2003.
- [2] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Processing Magazine*, vol. 20, pp. 18-29, 2003.
- [3] P. A. A. Assuncao and M. Ghanbari, "Frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 953-967, 1998.
- [4] J. Youn, M.-T. Sun, and J. Xin, "Video transcoder architectures for bit rate scaling of H.263 bit streams," presented at Proceedings of the 1999 7th International Multimedia Conference - ACM MULTIMEDIA '99, Oct 30-Nov 5 1999, Orlando, FL, USA, 1999.
- [5] P. N. Tudor and O. H. Werner, "Real-time transcoding of MPEG-2 video bit streams," presented at Proceedings of the 1997 International Broadcasting Convention, Sep 12-16 1997, Amsterdam, Neth, 1997.
- [6] ACTS, "Advanced television at low bit rates and networked transmission over integrated communication systems," *Online: web-site: <http://www.bbc.co.uk/atlantic/>*, 1996.
- [7] D. Lefol, D. Bull, N. Canagarajah, and F. Rovati, "Performance evaluation of transcoding algorithms for H.264," to be presented at ICCE 2006 - International Conference on Consumer Electronics, "in press", Las Vegas, USA, 2006.
- [8] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion estimation for high performance transcoding," *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 649-658, 1998.